ECON832 Final: Mini Paper

Daniel Sánchez Pazmiño

April 2024

In this mini paper, I summarize my findings from applying deep feedforward neural network models (DFNN) to predict choice between two lotteries in Plonsky et al. (2018) Choice Prediction Competition 2018 (CPC18). Two DFNss were trained, a baseline with risk features only and another with attention features.

1 Overview

The data comes from the CPC18 (Plonsky, Erev, and Ert 2017), which consists of experiments involving various decision makers choosing between two lotteries. While the complete dataset contains observations from Erev, Ert, and Plonsky (2017) 's Choice Prediction Competition 2015 (CPC2015), I focus on the newer data exclusive to CPC18 (Experiment 1), which added feedback and attention features to the CPC15 data.

This experiment involved sixty choice problems for 240 participants in two academic institutions in Israel. Two sets of problems were implemented, where participants faced one set of thirty problems. Each problem was faced for 25 trials, where the first five trials did not provide feedback about the forgone and obtained payoffs. A trial involved choosing between two options, where each option (A or B) involved an underlying lottery with a probability distribution (Plonsky et al. 2018). I use the calibration data set to train the DFNNs and the **individual track data** (Track II) to test the model.

2 Methodology

2.1 Data preparation

The calibration dataset was filtered to only include Experiment 1 data. The outcome variable was the binary indicator of the choice of B over A. Since the variable was already dichotomous, no transformation was needed. I include demographic variables in the baseline models:

location, gender and age. For location and gender, I dichotomized the variables with Technion and male as the reference levels.

Risk features which are considered include the shapes of the distributions for lotteries A and B. These involve the number of lottery outcomes, the expected values, the low payoffs, and probabilities to draw from the lotteries and lot distribution shapes. The latter defines the distribution of the lotteries, which can be binomial around the expected value, right skewed or left skewed. The shape of the lotteries where dichotomized in the same way as the location and gender variables; all other variables are continuous. The variance, while not directly observed, is a function of these variables.

Other relevant variables involve whether there is ambiguity in the probabilities or if there is aa correlation between the payoffs. Further, it is important to include trial numbers, potential and obtained payoffs obtained, and forgone payoffs. Some of these variables are categorical, but they are already dichotomized, so I didn't transform them.

Regarding attention features, I include the number of time block, reaction time, feedback binary outcome, the onscreen button and the trial number within a game.

The processing of the calibration dataset was done using `TidierData` (the Julia implementation of R's *dplyr*) and `DataFrames`. I performed a test-train split with 80% of the data used for training and 20% for testing. I further tested with the individual track data, which was not used in the training process.

2.2 Feature engineering

As mentioned before, all categorical variables for both the risk and attention features were dichotomized. This is equivalent to **one-hot encoding**. The continuous variables were standardized to have a mean of zero and a standard deviation of one.

The data was processed from its rectangular form (every row per decision) to a long format where each row represented an attribute, and every column represented a decision. This is the standard format for neural networks using `Flux`.

2.3 Model configuration

The model was created using the `Flux` which composes multiple layers together to form a more complex network. The layers are executed in the order they are defined.

The first layer is a `Dense` layer with the number of input nodes that are fed as features, 64 neurons, and a Rectified Linear Unit (ReLU) activation function for the input and hidden layer. The output layer is a `Dense` layer with 1 output node and a sigmoid activation function. The sigmoid function is commonly used for binary classification problems, as it squashes its input values to be between 0 and 1, which can be interpreted as probabilities.

I use mean squared error as the loss function and the ADAM optimizer to minimize the loss function. The learning rate was set to 0.5.

3 Feature analysis

References

Erev, Ido, Eyal Ert, and Ori Plonsky. 2017. "Raw Data for CPC2015: A Choice Prediction Competition for Decisions Under Risk, Under Ambiguity, and from Experience." Data set with codebook. Zenodo. https://doi.org/10.5281/zenodo.321652.

Plonsky, Ori, Reut Apel, Ido Erev, Eyal Ert, and Moshe Tennenholtz. 2018. "When and How Can Social Scientists Add Value to Data Scientists? A Choice Prediction Competition for Human Decision Making." *Unpublished Manuscript*. https://cpc-18.com/wp-content/uploads/2018/03/cpc18-white-paper-march-update.pdf.

Plonsky, Ori, Ido Erev, and Eyal Ert. 2017. "Calibration Data for Choice Prediction Competition 2018 (CPC18)." Data set with codebook. Zenodo. https://doi.org/10.5281/zenodo.845873.